Dement Neurocogn Disord. 2018 Sep;17(3):83-89
https://doi.org/10.12779/dnd.2018.17.3.83
pISSN 1738-1495·eISSN 2384-0757

DND Dementia and Neurocognitive Disorder

Review Article

Check for updates

Artificial Neural Network: Understanding the Basic Concepts without Mathematics

Su-Hyun Han,[1] **Ko Woon Kim**,[2] **SangYun Kim**,[3] **Young Chul Youn**[1]

[1]Department of Neurology, Chung-Ang University College of Medicine, Seoul, Korea
[2]Department of Neurology, Chonbuk National University Hospital, Jeonju, Korea
[3]Department of Neurology, Seoul National University College of Medicine and Seoul National University Bundang Hospital, Seongnam, Korea

Received: Sep 30, 2018
Revised: Nov 4, 2018
Accepted: Nov 20, 2018

Correspondence to
Young Chul Youn, MD, PhD
Department of Neurology, Chung-Ang University College of Medicine, Chung-Ang University Hospital, 102 Heukseok-ro, Dongjak-gu, Seoul 06973, Korea.
E-mail: neudoc@cau.ac.kr

ORCID iDs
Su-Hyun Han
https://orcid.org/0000-0002-3084-6985
SangYun Kim
https://orcid.org/0000-0002-9101-5704
Young Chul Youn
https://orcid.org/0000-0002-2742-1759

Funding
This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2017S1A6A3A01078538).

Conflict of Interest
The authors have no financial conflicts of interest.

ABSTRACT

Machine learning is where a machine (i.e., computer) determines for itself how input data is processed and predicts outcomes when provided with new data. An artificial neural network is a machine learning algorithm based on the concept of a human neuron. The purpose of this review is to explain the fundamental concepts of artificial neural networks.

Keywords: Machine Learning; Artificial Intelligence; Neural Networks; Deep Learning

WHAT IS THE DIFFERENCE BETWEEN MACHINE LEARNING AND A COMPUTER PROGRAM?

A computer program takes input data, processes the data, and outputs a result. A programmer stipulates how the input data should be processed. In machine learning, input and output data is provided, and the machine determines the process by which the given input produces the given output data.[1-3] This process can then predict the unknown output when new input data is provided.[1-3]

So, how does machine determine the process? Consider a linear model, $y = Wx + b$. If we are given the x and y values shown in **Table 1**, we know intuitively that $W = 2$ and $b = 0$.

However, how does a computer know what W and b are? W and b are randomly generated initially. For example, if we initially say $W = 1$ and $b = 0$ and input the x values from **Table 1**, the

Table 1. Example dataset for the linear model $y = Wx + b$

x	y
1	2
2	4
3	6
4	8
5	10
10	20
20	40
30	60

Author Contributions
Conceptualization: Kim SY; Supervision: Kim KW, Youn YC; Writing - original draft: Han SH, Kim KW; Writing - review & editing: Youn YC.

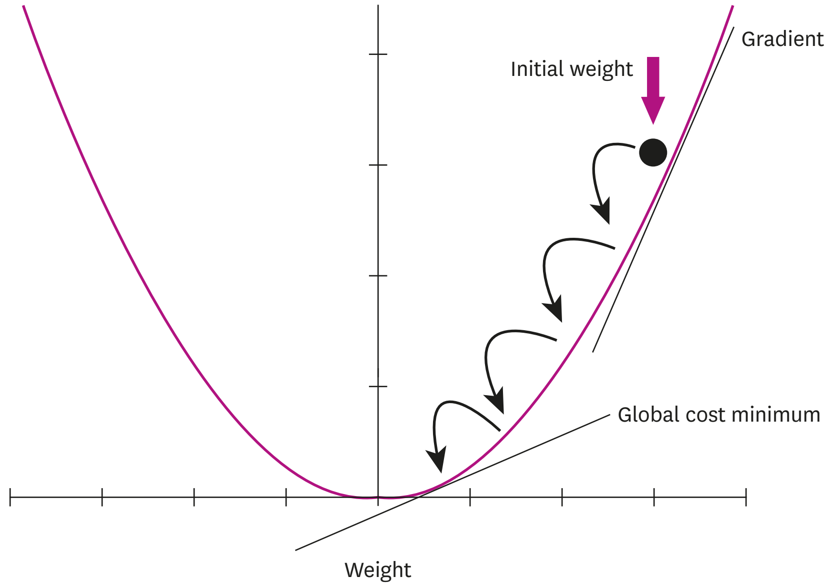

Fig. 1. A graph of a cost function (modified from https://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/).

predicted *y* values do not match the *y* values in the **Table 1**. At this point, there is a difference between the correct *y* values and the predicted *y* values; the machine gradually adjusts the values of *W* and *b* to reduce this difference. The difference between the predicted values and the correct values is called the cost function. Minimizing the cost function makes predictions closer to the correct answers.[4-7]

The costs that correspond to given *W* and *b* values are shown in the **Fig. 1** (obtained from https://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/).[8-10] Given the current values of *W* and *b*, the gradient is obtained. If the gradient is positive, the values of *W* and *b* are decreased. If the gradient is negative, the values of *W* and *b* are increased. In other words, the values of *W* and *b* determined by the derivative of the cost function.[8-10]

ARTIFICIAL NEURAL NETWORK

The basic unit by which the brain works is a neuron. Neurons transmit electrical signals (action potentials) from one end to the other.[11] That is, electrical signals are transmitted from the dendrites to the axon terminals through the axon body. In this way, the electrical signals continue to be transmitted across the synapse from one neuron to another. The human brain has approximately 100 billion neurons.[11,12] It is difficult to imitate this level of complexity with existing computers. However, Drosophila have approximately 100,000 neurons, and they are able to find food, avoid danger, survive, and reproduce.[13] Nematodes have 302 neurons and they survive well.[14] This is a level of complexity that can be replicated well even with today's computers. However, nematodes can perform much better than our computers.

Let's think of the operative principles of neurons. Neurons receive signals and generate other signals. That is, they receive input data, perform some processing, and give an output.[11,12] However, the output is not given at a constant rate; the output is generated when the input

DND Dementia and Neurocognitive Disorder

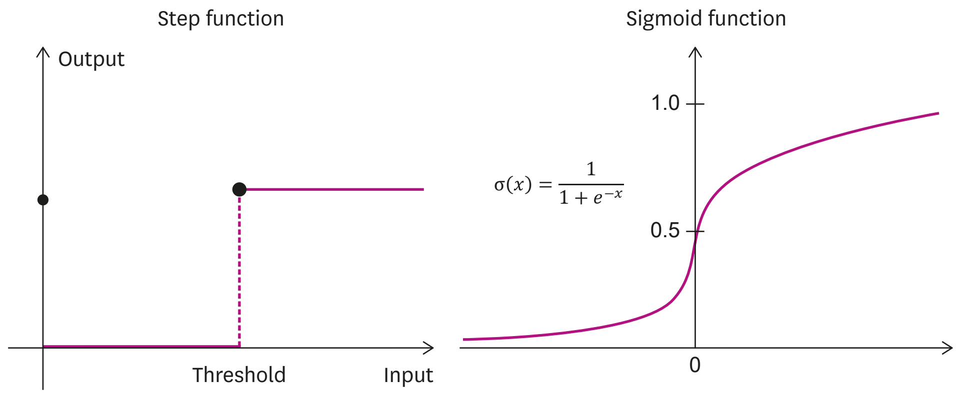

Fig. 2. A step function and a sigmoid function.

exceeds a certain threshold.[15] The function that receives an input signal and produces an output signal after a certain threshold value is called an activation function.[11,15] As shown in **Fig. 2**, when the input value is small, the output value is 0, and once the input value rises above the threshold value, a non-zero output value suddenly appears. Thus, the responses of biological neurons and artificial neurons (nodes) are similar. However, in reality, artificial neural networks use various functions other than activation functions, and most of them use sigmoid functions.[4,6] which are also called logistic functions. The sigmoid function has the advantage that it is very simple to calculate compared to other functions. Currently, artificial neural networks predominantly use a weight modification method in the learning process.[4,5,7] In the course of modifying the weights, the entire layer requires an activation function that can be differentiated. This is because the step function cannot be used as it is. The sigmoid function is expressed as the following equation.[6,16]

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Biological neurons receive multiple inputs from pre-synaptic neurons.[11] Neurons in artificial neural networks (nodes) also receive multiple inputs, then they add them and process the sum with a sigmoid function.[5,7] The value processed by the sigmoid function then becomes the output value. As shown in **Fig. 3**, if the sum of inputs A, B, and C exceeds the threshold and the sigmoid function works, this neuron generates an output value.[4]

A neuron receives input from multiple neurons and transmits signals to multiple neurons.[4,5]

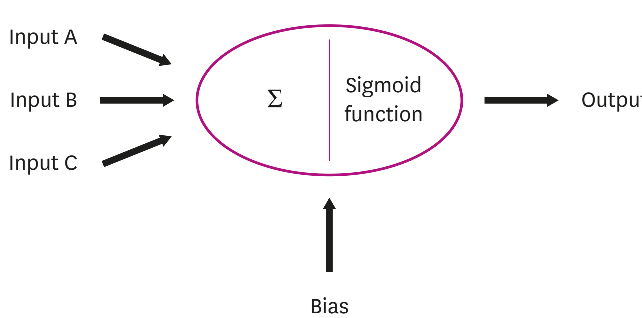

Fig. 3. Input and output of information from neurons.

https://doi.org/10.12779/dnd.2018.17.3.83

DND Dementia and Neurocognitive Disorder

The models of biological neurons and an artificial neural network are shown in **Fig. 4**. Neurons are located over several layers, and one neuron is considered to be connected to multiple neurons. In an artificial neural network, the first layer (input layer) has input neurons that transfer data via synapses to the second layer (hidden layer), and similarly, the hidden layer transfers this data to the third layer (output layer) via more synapses.[4,5,7] The hidden layer (node) is called the "black box" because we are unable to interpret how an artificial neural network derived a particular result.[4-6]

So, how does a neural network learn in this structure? There are some variables that must be updated in order to increase the accuracy of the output values during the learning process.[4,5,7] At this time, there could be a method of adjusting the sum of the input values or modifying the sigmoid function, but it is a rather practical method to adjust the connection strength between the neurons (nodes).[4,5,7] **Fig. 5** is a reformulation of **Fig. 4** with the weight to be applied to the connection strength. Low weights weaken the signal, and high weights enhance the signal.[17] The learning part is the process by which the network adjusts the weights to improve the output.[4,5,7] The weights *W*1,2, *W*1,3, *W*2,3, *W*2,2, and *W*3,2 are

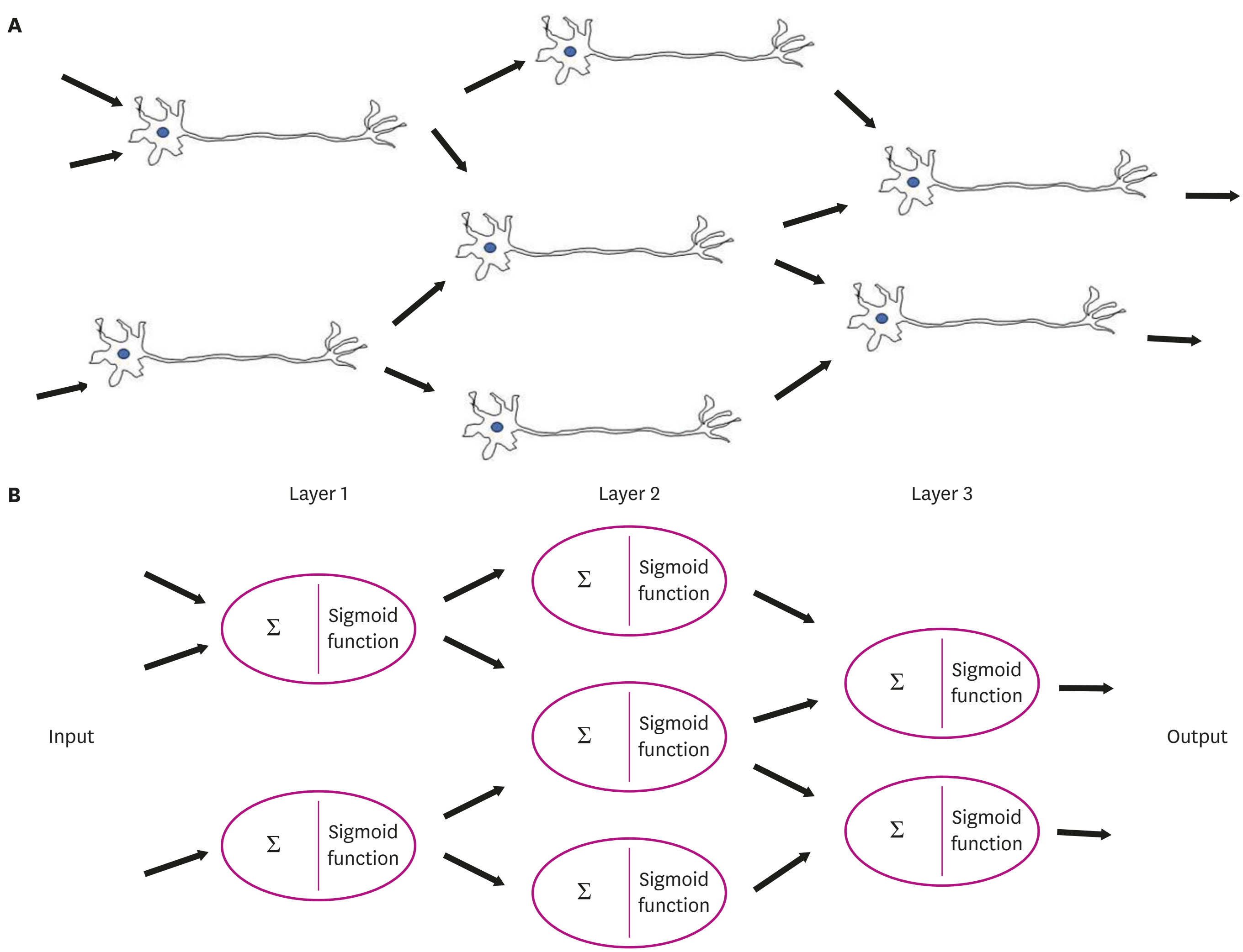

Fig. 4. (A) Biological neural network and (B) multi-layer perception in an artificial neural network.

https://doi.org/10.12779/dnd.2018.17.3.83

DND Dementia and Neurocognitive Disorder

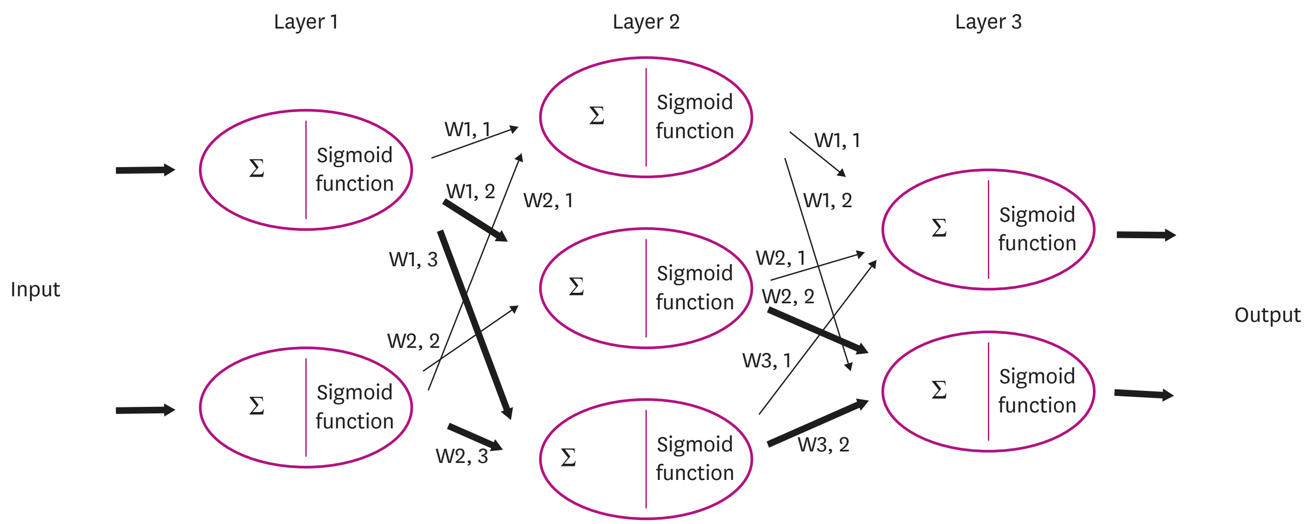

Fig. 5. The connections and weights between neurons of each layer in an artificial neural network.

emphasized by strengthening the signal due to the high weight. If the weight W is 0, the signal is not transmitted, and the network cannot be influenced.

Multiple nodes and connections could affect predicted values and their errors. In this case, how do we update the weights to get the correct output, and how does learning work?

The updating of the weights is determined by the error between the predicted output and the correct output.[4,5,7] The error is divided by the ratio of the weights on the links, and the divided errors are back propagated and reassembled (**Fig. 6**). However, in a hierarchical structure, it is extremely difficult to calculate all the weights mathematically. As an alternative, we can use the gradient descent method[8-10] to reach the correct answer — even if we do not know the complex mathematical calculations (**Fig. 1**). The gradient descent method is a technique to find the lowest point at which the cost is minimized in a cost function (the difference between the predicted value and the answer obtained by the arbitrarily start weight W). Again, the machine can start with any W value and alter it gradually (so that it goes down the graph) so that the cost is reduced and finally reaches a minimum. Without complicated mathematical calculations, this minimizes the error between the predicted value and the

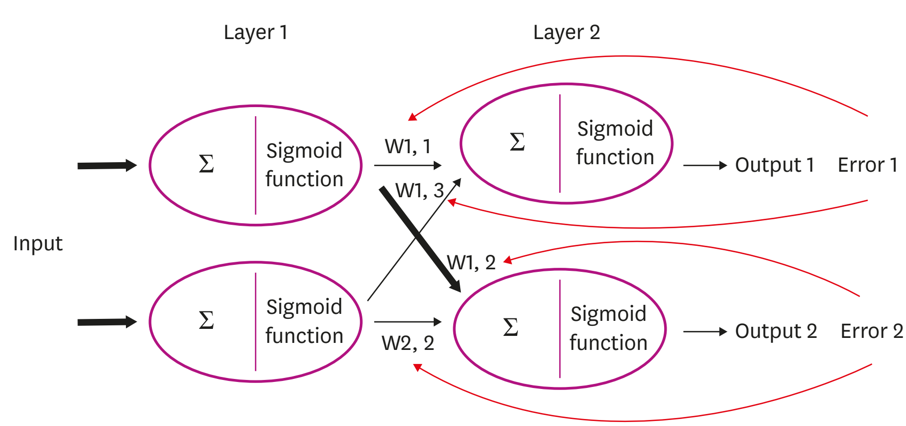

Fig. 6. Back propagation of error for updating the weights.

https://doi.org/10.12779/dnd.2018.17.3.83

answer (or it shows that the arbitrary start weight is correct). This is the learning process of artificial neural networks.

CONCLUSIONS

In conclusion, the learning process of an artificial neural network involves updating the connection strength (weight) of a node (neuron). By using the error between the predicted value and the correct, the weight in the network is adjusted so that the error is minimized and an output close to the truth is obtained.

REFERENCES

1. Samuel AL. Some studies in machine learning using the game of checkers. IBM J Res Develop 1959;3:210-229.
CROSSREF
2. Koza JR, Bennett FH 3rd, Andre D, Keane MA. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In: Gero JS, Sudweeks F, editors. *Artificial Intelligence in Design*. Dordrecht: Kluwer Academic, 1996;151-170.
3. Deo RC. Machine learning in medicine. Circulation 2015;132:1920-1930.
PUBMED | CROSSREF
4. Rashid T. *Make Your Own Neural Network*. 1st ed. North Charleston, SC: CreateSpace Independent Publishing Platform, 2016.
5. Haykin S, Haykin SS. *Neural Networks and Learning Machines*. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2009.
6. Rashid T, Huang BQ, Kechadi MT. A new simple recurrent network with real-time recurrent learning process. Proceedings of the 14th Irish Conference on Artificial Intelligence & Cognitive Science, AICS 2003; 2003 September 17–19; Dublin, Ireland. [place unknown]: Artificial Intelligence and Cognitive Science, 2003;169-174.
7. Zakaria M, AL-Shebany M, Sarhan S. Artificial neural network: a brief overview. Int J Eng Res Appl 2014;4:7-12.
8. Bottou L. Chapter 2. On-line learning and stochastic approximations. In: Saad D, editor. *On-Line Learning in Neural Networks*. Cambridge: Cambridge University Press, 1999;9-42.
CROSSREF
9. Bottou L. Large-scale machine learning with stochastic gradient descent. In: Lechevallier Y, Saporta G, editors. Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics; 2010 August 22–27; Paris, France. Heidelberg: Physica-Verlag HD, 2010;177-186.
CROSSREF
10. Bottou L. Stochastic gradient descent tricks. In: Montavon G, Orr GB, Müller KR, editors. *Neural Networks: Tricks of the Trade*. 2nd ed. Heidelberg: Springer-Verlag Berlin Heidelberg, 2012;421-436.
CROSSREF
11. Ebersole JS, Husain AM, Nordli DR. *Current Practice of Clinical Electroencephalography*. 4th ed. Philadelphia, PA: Wolters Kluwer, 2014.
12. Shepherd GM, Koch C. Introduction to synaptic circuits. In: Shepherd GM, editor. *The Synaptic Organization of the Brain*. 3rd ed. New York, NY: Oxford University Press, 1990;3-31.
13. Alivisatos AP, Chun M, Church GM, Greenspan RJ, Roukes ML, Yuste R. The brain activity map project and the challenge of functional connectomics. Neuron 2012;74:970-974.
PUBMED | CROSSREF
14. Hobert O. Neurogenesis in the nematode *Caenorhabditis elegans*. WormBook 2010:1-24.
PUBMED | CROSSREF
15. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. 1943. Bull Math Biol 1990;52:99-115.
PUBMED | CROSSREF

16. Ranka S, Mohan CK, Mehrotra K, Menon A. Characterization of a class of sigmoid functions with applications to neural networks. Neural Netw 1996;9:819-835.
PUBMED | CROSSREF
17. Kozyrev SV. Classification by ensembles of neural networks. p-Adic Numbers Ultrametric Anal Appl 2012;4:27-33.
CROSSREF